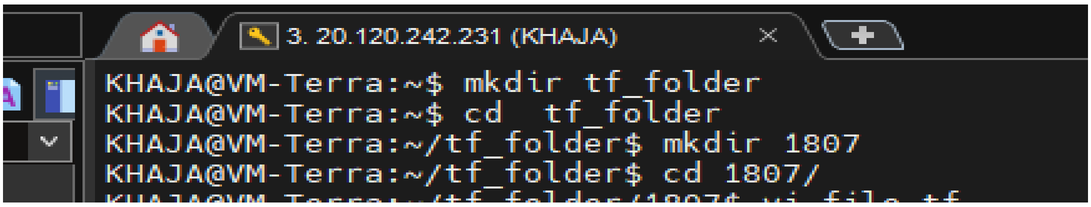
Terraform Task

Name:Shaik Khaja Basha
Batch : Batch 11
Date : 18.07.2025
Task : File creations in .tf & make corrections

1. **make a mistake in configuration->run validate->correct and execute for that i had like abc.txt file in terraform**

Ans: Open your terminal and run:

```
mkdir tf_folder        # Main folder for Terraform work
cd tf_folder          # Move into that folder
mkdir 1807             # Create a sub-folder for today's date or task
cd 1807               # Move into that sub-folder
```



```
KHAJA@VM-Terra:~$ mkdir tf_folder
KHAJA@VM-Terra:~$ cd  tf_folder
KHAJA@VM-Terra:~/tf_folder$ mkdir 1807
KHAJA@VM-Terra:~/tf_folder$ cd 1807/
KHAJA@VM-Terra:~/tf_folder/1807$ vi file.tf
```

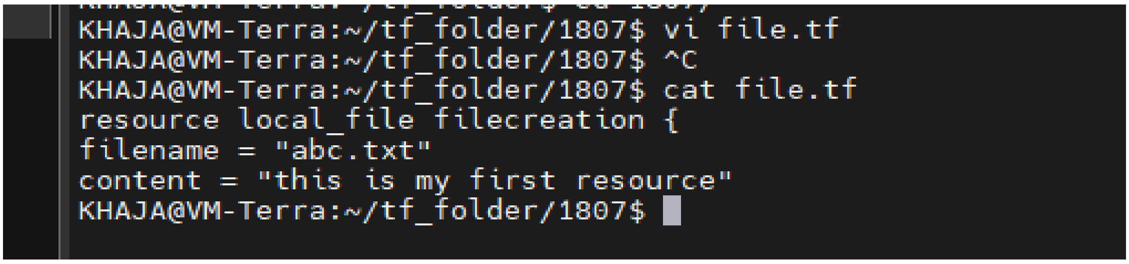
- ❖ Create Terraform configuration file
  - ○ vi file.tf
  - ○ In vi, press i to enter **insert mode**, then **paste this content**:

Syntax : resource "local_file" "filecreation" {

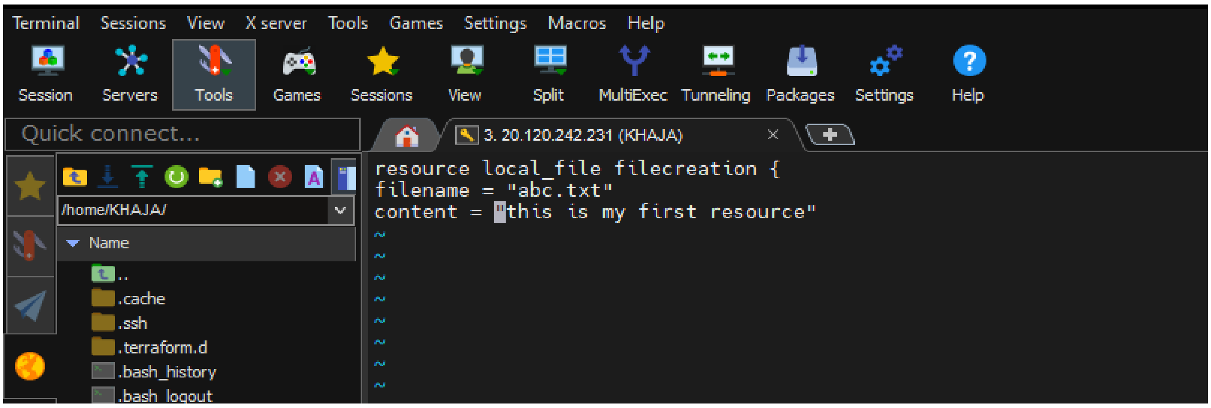
filename = "abc.txt"

content = "this is my first resource"

}



```
KHAJA@VM-Terra:~/tf_folder$ cd 1807/
KHAJA@VM-Terra:~/tf_folder/1807$ vi file.tf
KHAJA@VM-Terra:~/tf_folder/1807$ ^C
KHAJA@VM-Terra:~/tf_folder/1807$ cat file.tf
resource local_file filecreation {
filename = "abc.txt"
content = "this is my first resource"
KHAJA@VM-Terra:~/tf_folder/1807$
```



```
resource local_file filecreation {
filename = "abc.txt"
content = "this is my first resource"
~
~
~
~
~
~
~
~
```

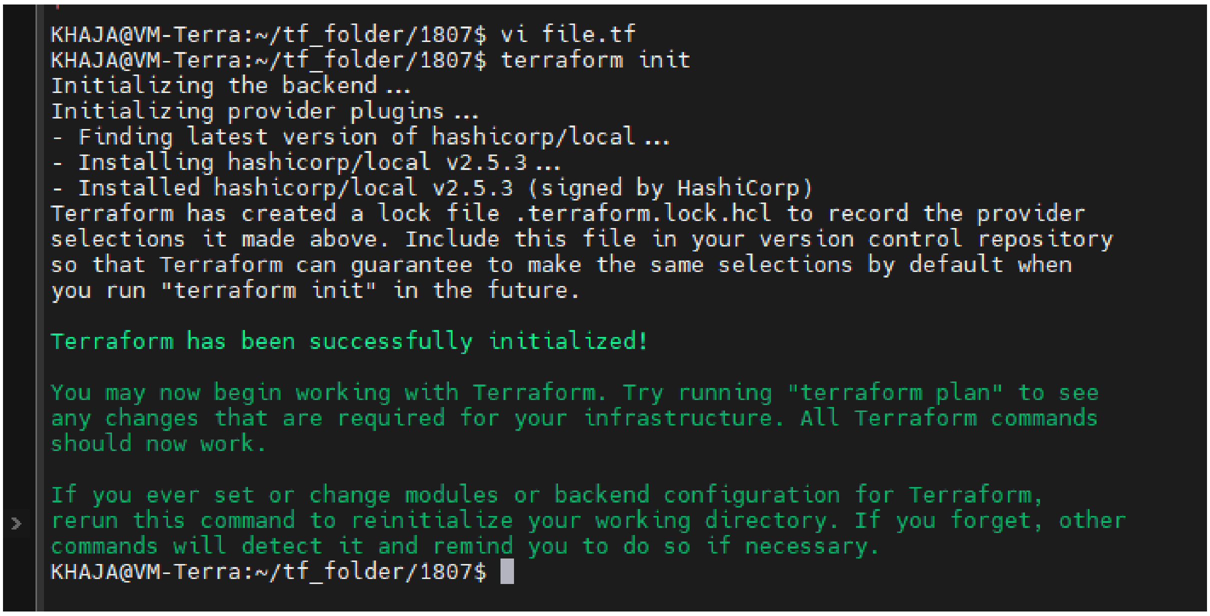
- Press Esc
- Type :wq and hit **Enter** to save and exit

Initialize Terraform

- terraform init

This will download the required local provider plugin and set up .terraform directory.

Output will look like:



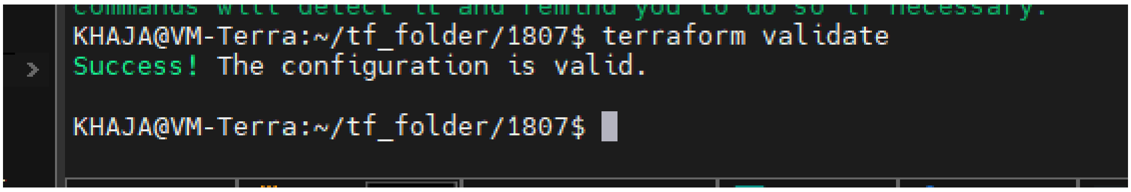
```
KHAJA@VM-Terra:~/tf_folder/1807$ vi file.tf
KHAJA@VM-Terra:~/tf_folder/1807$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
KHAJA@VM-Terra:~/tf_folder/1807$
```

Validate the configuration:

Check if your configuration is correct:

- terraform validate

output:

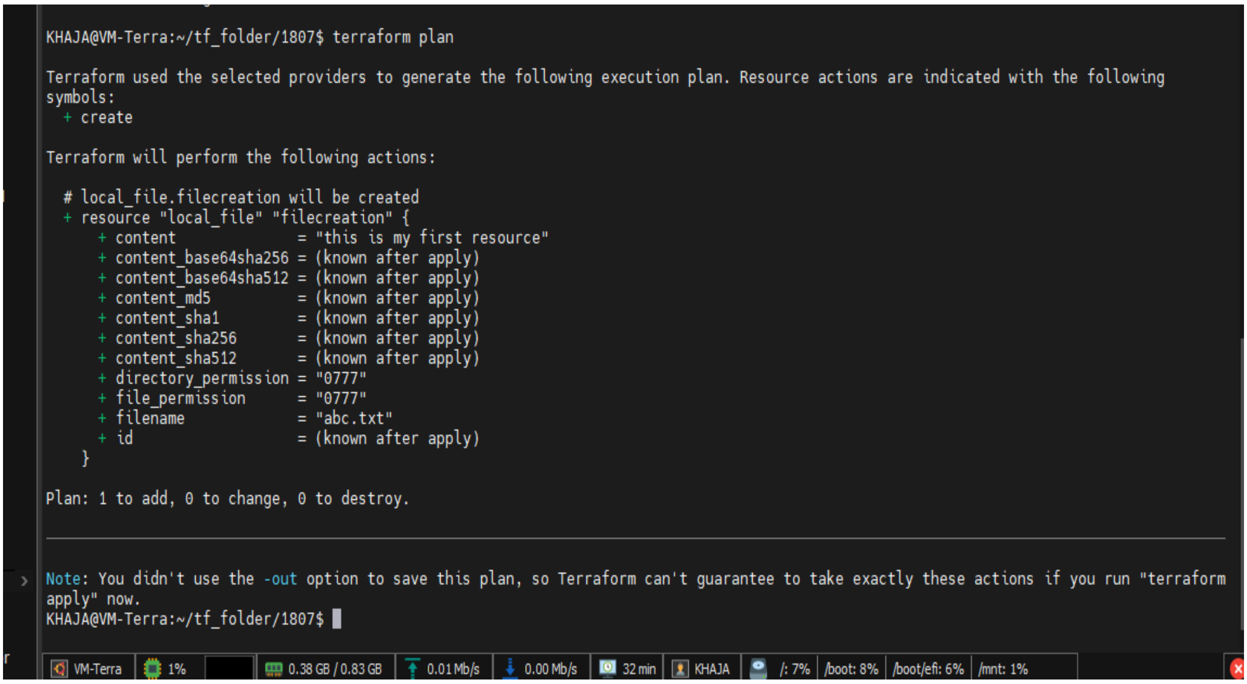


```
commands will detect it and remind you to do so if necessary.
KHAJA@VM-Terra:~/tf_folder/1807$ terraform validate
Success! The configuration is valid.

KHAJA@VM-Terra:~/tf_folder/1807$
```

Its show me as validate “Success”

- ❖ **Plan the Infrastructure**
  - ➢ terraform plan

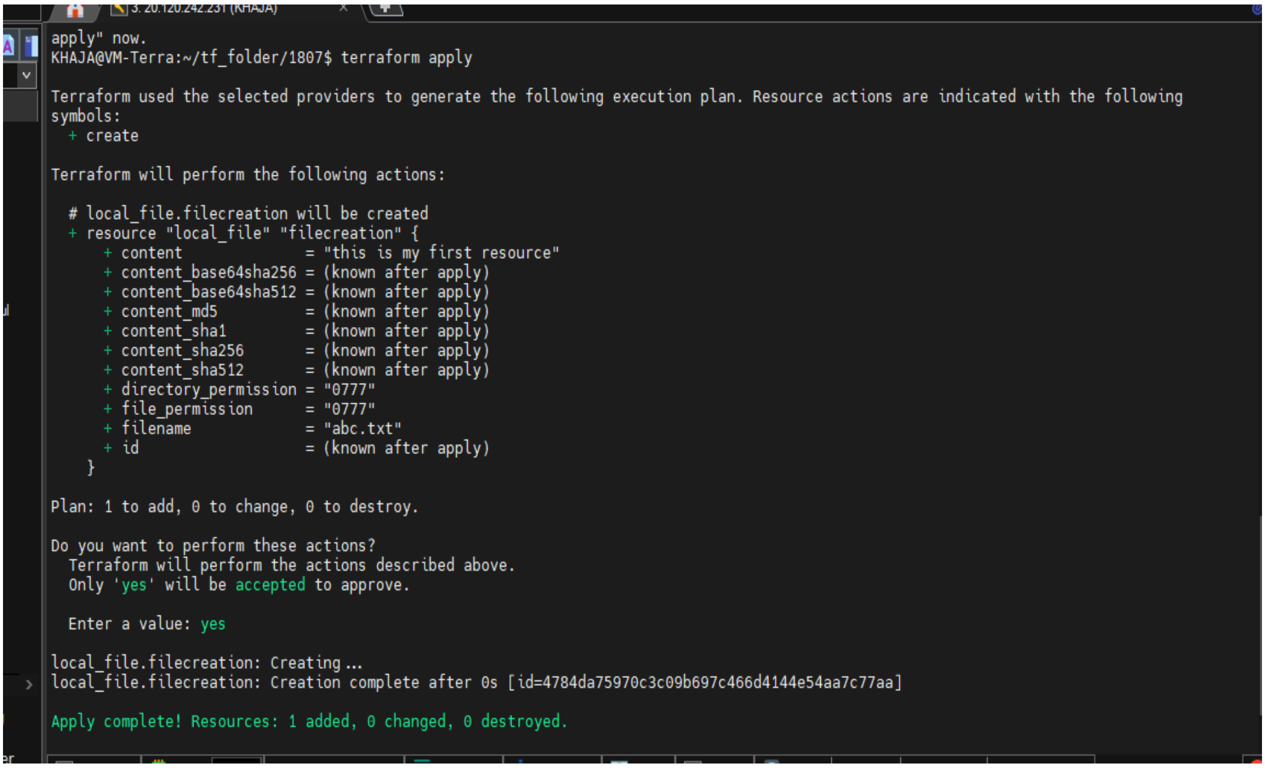
This step shows what Terraform **will do** without actually doing it.



```
KHAJA@VM-Terra:~/tf_folder/1807$ terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

  # local_file.filecreation will be created
  + resource "local_file" "filecreation" {
      + content              = "this is my first resource"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "abc.txt"
      + id                   = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform
apply" now.
KHAJA@VM-Terra:~/tf_folder/1807$
```

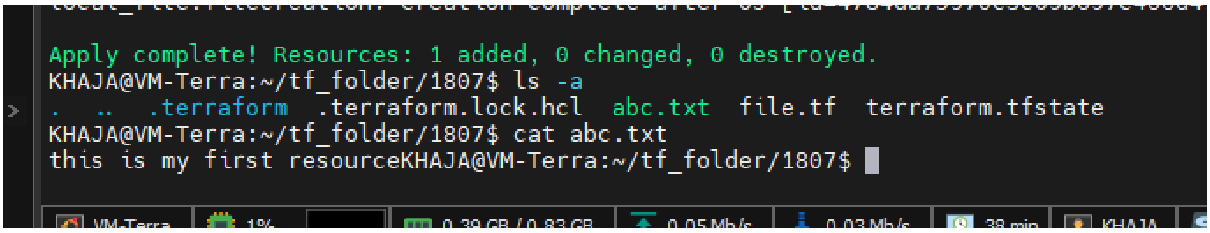
- ❖ **Apply the Configuration**
  - ➢ terraform apply

This step actually **creates the file**.



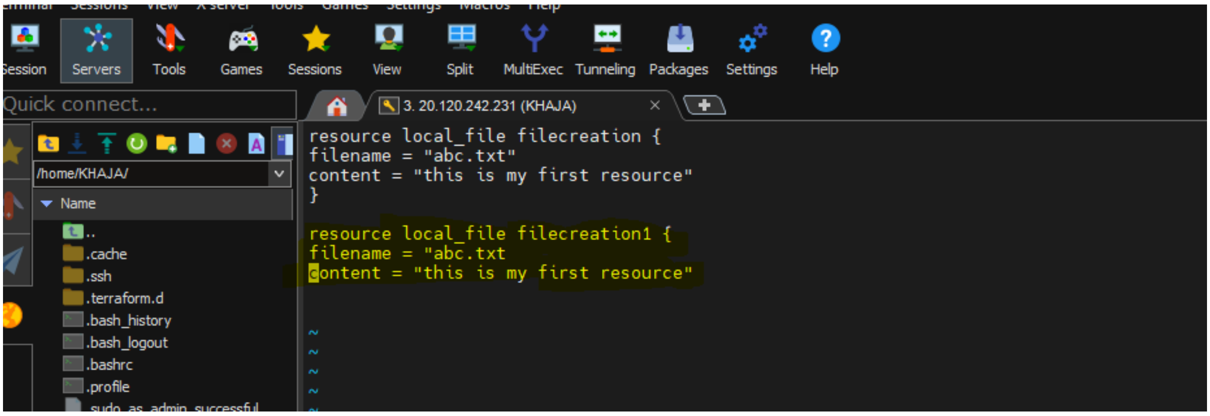
```
apply" now.
KHAJA@VM-Terra:~/tf_folder/1807$ terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

  # local_file.filecreation will be created
  + resource "local_file" "filecreation" {
      + content              = "this is my first resource"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "abc.txt"
      + id                   = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.filecreation: Creating...
local_file.filecreation: Creation complete after 0s [id=4784da75970c3c09b697c466d4144e54aa7c77aa]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

Verify Output



```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
KHAJA@VM-Terra:~/tf_folder/1807$ ls -a
.  ..  .terraform  .terraform.lock.hcl  abc.txt  file.tf  terraform.tfstate
KHAJA@VM-Terra:~/tf_folder/1807$ cat abc.txt
this is my first resourceKHAJA@VM-Terra:~/tf_folder/1807$ 
```

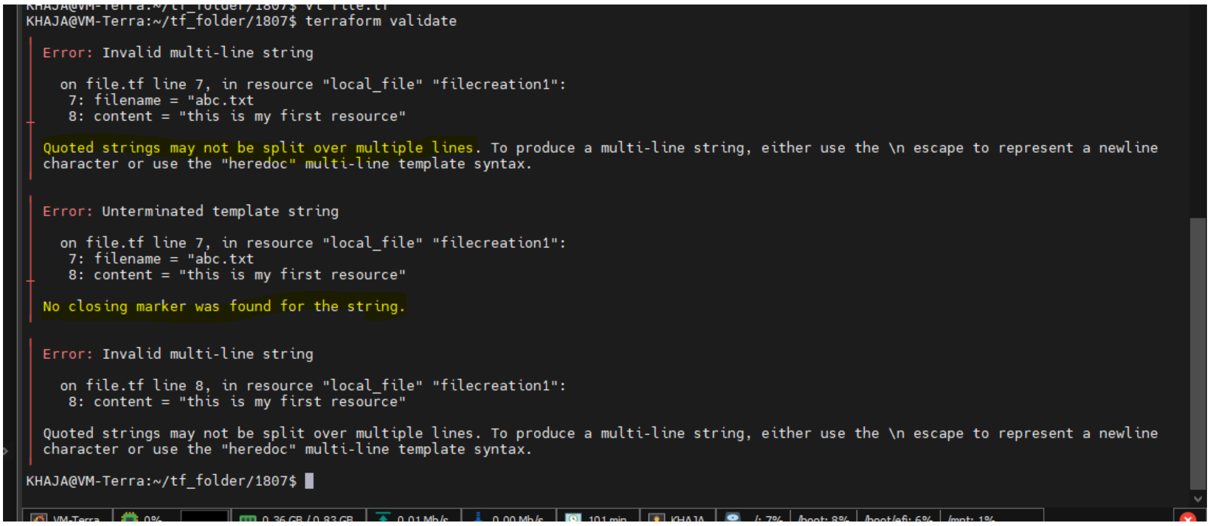
Add Terraform code with a mistake (Missing closing brace)



```
resource local_file filecreation {
filename = "abc.txt"
content = "this is my first resource"
}

resource local_file filecreation1 {
filename = "abc.txt
content = "this is my first resource"
```

When I create a another resource within the same file but it show the error

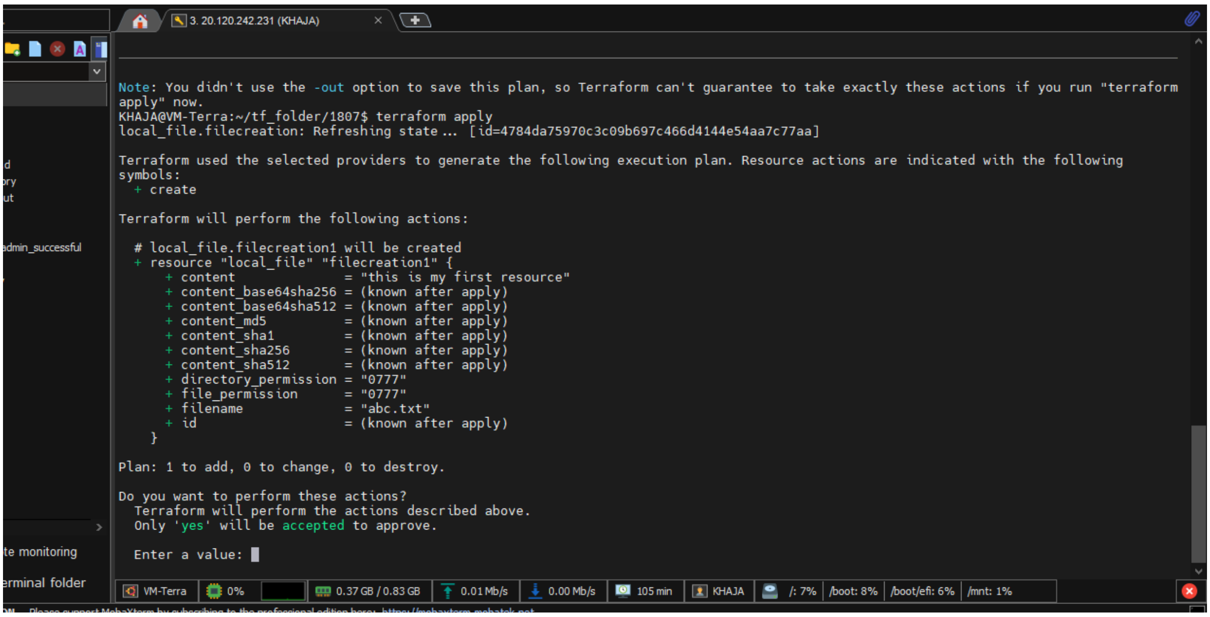
- ➢ Terraform validate



```
KHAJA@VM-Terra:~/tf_folder/1807$ vi file.tf
KHAJA@VM-Terra:~/tf_folder/1807$ terraform validate

  Error: Invalid multi-line string

    on file.tf line 7, in resource "local_file" "filecreation1":
     7: filename = "abc.txt
     8: content = "this is my first resource"

  Quoted strings may not be split over multiple lines. To produce a multi-line string, either use the \n escape to represent a newline
  character or use the "heredoc" multi-line template syntax.


  Error: Unterminated template string

    on file.tf line 7, in resource "local_file" "filecreation1":
     7: filename = "abc.txt
     8: content = "this is my first resource"

  No closing marker was found for the string.


  Error: Invalid multi-line string

    on file.tf line 8, in resource "local_file" "filecreation1":
     8: content = "this is my first resource"

  Quoted strings may not be split over multiple lines. To produce a multi-line string, either use the \n escape to represent a newline
  character or use the "heredoc" multi-line template syntax.

KHAJA@VM-Terra:~/tf_folder/1807$ 
```

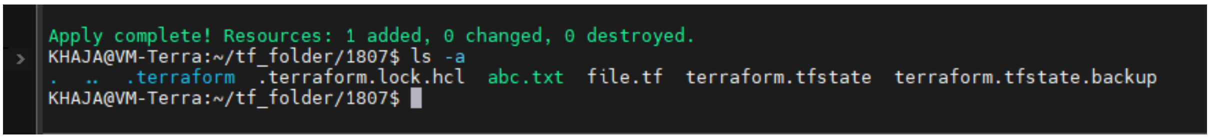
Lets correct the configuration and make execute

- ➢ Terraform apply



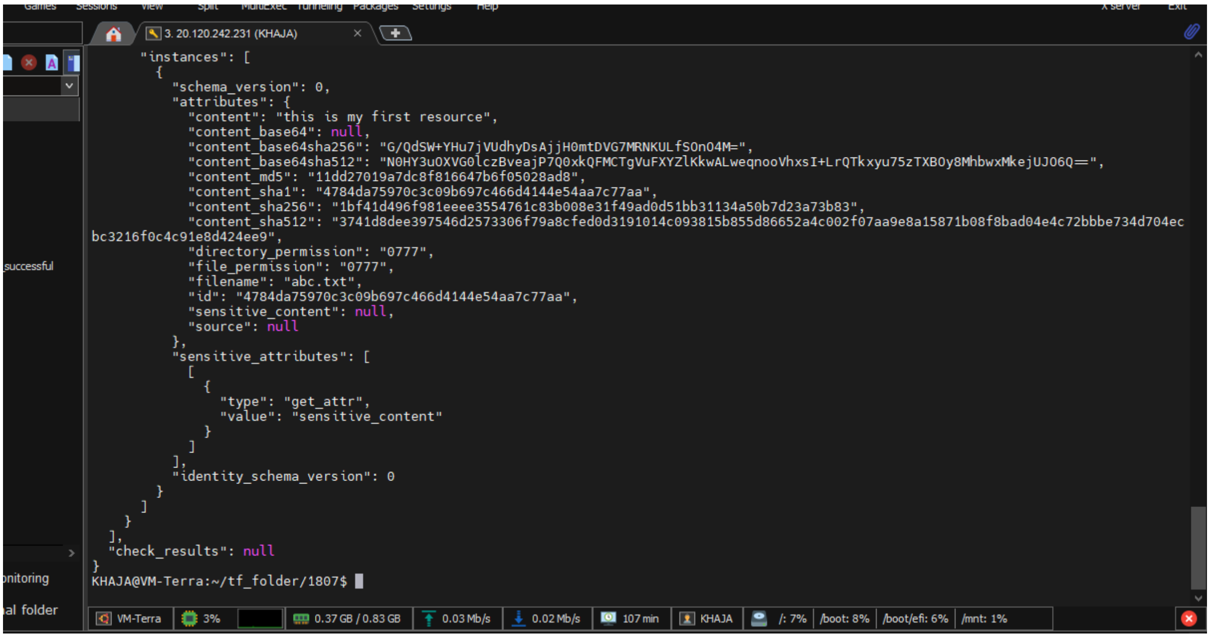
```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform
apply" now.
KHAJA@VM-Terra:~/tf_folder/1807$ terraform apply
local_file.filecreation: Refreshing state... [id=4784da75970c3c09b697c466d4144e54aa7c77aa]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

  # local_file.filecreation1 will be created
  + resource "local_file" "filecreation1" {
      + content              = "this is my first resource"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "abc.txt"
      + id                   = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: 
```

Know it will create a file with no error



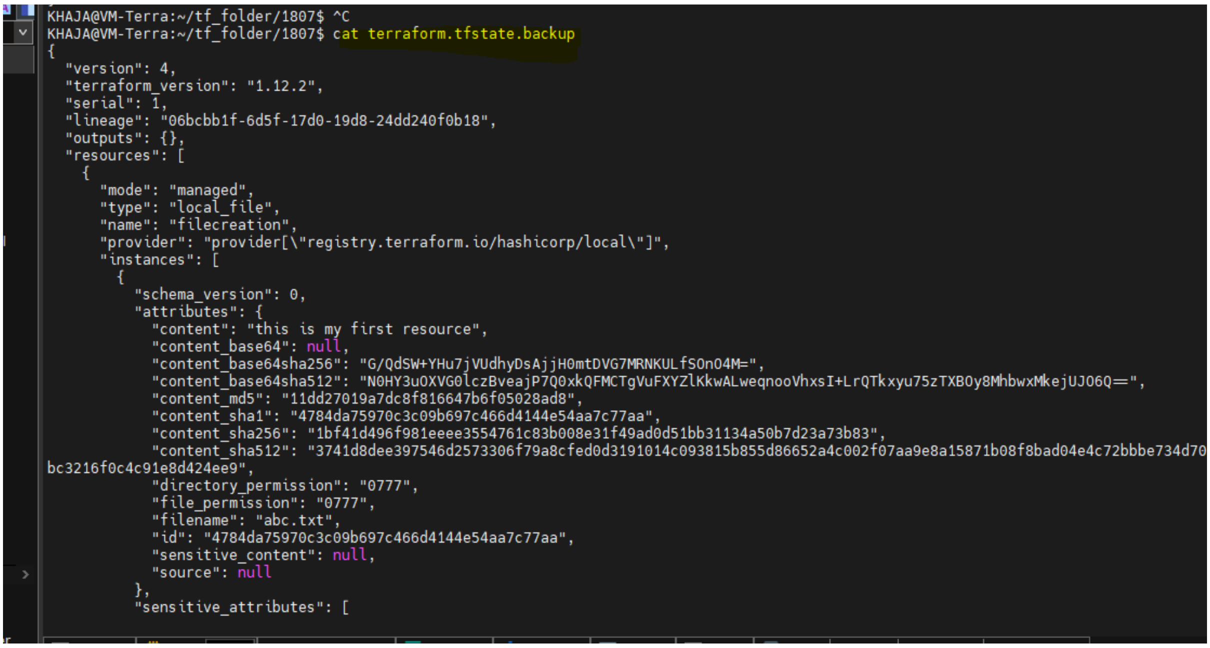
```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
KHAJA@VM-Terra:~/tf_folder/1807$ ls -a
.  ..  .terraform  .terraform.lock.hcl  abc.txt  file.tf  terraform.tfstate  terraform.tfstate.backup
KHAJA@VM-Terra:~/tf_folder/1807$ 
```

When we check the state file it will show the previous execution



```
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "content": "this is my first resource",
            "content_base64": null,
            "content_base64sha256": "G/QdSW+YHu7jVUdhyDsAjjH0mtDVG7MRNKULfSOnO4M=",
            "content_base64sha512": "N0HY3uOXVG0lczBveajP7Q0xkQFMCTgVuFXYZlKkwALweqnooVhxsI+LrQTkxyu75zTXBOy8MhbwxMkejUJ06Q==",
            "content_md5": "11dd27019a7dc8f816647b6f05028ad8",
            "content_sha1": "4784da75970c3c09b697c466d4144e54aa7c77aa",
            "content_sha256": "1bf41d496f981eeee3554761c83b008e31f49ad0d51bb31134a50b7d23a73b83",
            "content_sha512": "3741d8dee397546d2573306f79a8cfed0d3191014c093815b855d86652a4c002f07aa9e8a15871b08f8bad04e4c72bbbe734d704ec
bc3216f0c4c91e8d424ee9",
            "directory_permission": "0777",
            "file_permission": "0777",
            "filename": "abc.txt",
            "id": "4784da75970c3c09b697c466d4144e54aa7c77aa",
            "sensitive_content": null,
            "source": null
          },
          "sensitive_attributes": [
            [
              {
                "type": "get_attr",
                "value": "sensitive_content"
              }
            ]
          ],
          "identity_schema_version": 0
        }
      ]
    }
  ],
  "check_results": null
}
KHAJA@VM-Terra:~/tf_folder/1807$ 
```

For the backup file it contain the previous execution



```
KHAJA@VM-Terra:~/tf_folder/1807$ ^C
KHAJA@VM-Terra:~/tf_folder/1807$ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.12.2",
  "serial": 1,
  "lineage": "06bcbb1f-6d5f-17d0-19d8-24dd240f0b18",
  "outputs": {},
  "resources": [
    {
      "mode": "managed",
      "type": "local_file",
      "name": "filecreation",
      "provider": "provider[\"registry.terraform.io/hashicorp/local\"]",
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "content": "this is my first resource",
            "content_base64": null,
            "content_base64sha256": "G/QdSW+YHu7jVUdhyDsAjjH0mtDVG7MRNKULfSOnO4M=",
            "content_base64sha512": "N0HY3uOXVG0lczBveajP7Q0xkQFMCTgVuFXYZlKkwALweqnooVhxsI+LrQTkxyu75zTXBOy8MhbwxMkejUJO6Q==",
            "content_md5": "11dd27019a7dc8f816647b6f05028ad8",
            "content_sha1": "4784da75970c3c09b697c466d4144e54aa7c77aa",
            "content_sha256": "1bf41d496f981eeee3554761c83b008e31f49ad0d51bb31134a50b7d23a73b83",
            "content_sha512": "3741d8dee397546d2573306f79a8cfed0d3191014c093815b855d86652a4c002f07aa9e8a15871b08f8bad04e4c72bbbe734d70
bc3216f0c4c91e8d424ee9",
            "directory_permission": "0777",
            "file_permission": "0777",
            "filename": "abc.txt",
            "id": "4784da75970c3c09b697c466d4144e54aa7c77aa",
            "sensitive_content": null,
            "source": null
          },
          "sensitive_attributes": [
```

2. create 3 .tf file with single resources inside

Ans: three Terraform (.tf) files, each containing a single resource:

Creating files

Create a file name file1.tf using vi file1.tf

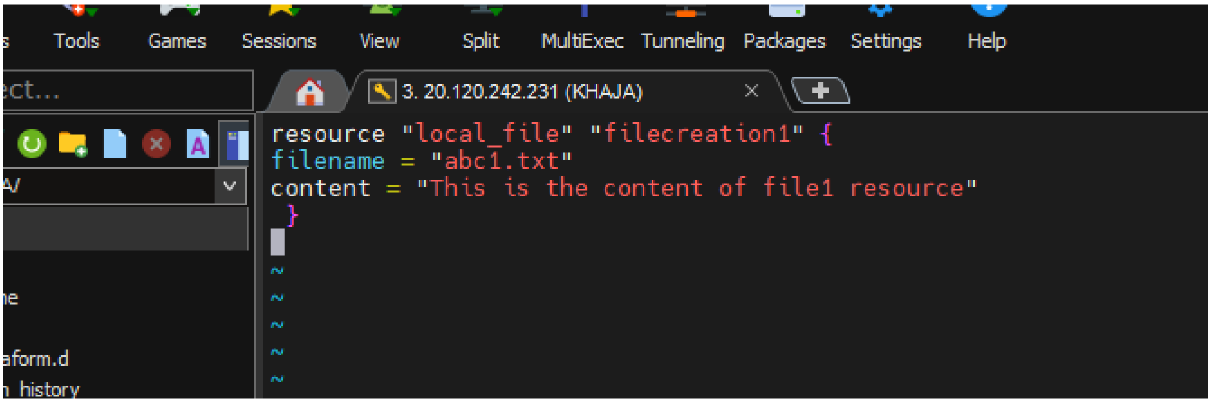
Enter the resource in file1.tf and save it esc:wq

resource "local_file" "filecreation1" {

filename = "abc1.txt"

content = "This is the content of file1 resource"

}



```
resource "local_file" "filecreation1" {
filename = "abc1.txt"
content = "This is the content of file1 resource"
 }

~
~
~
~
~
```

Know we have created file1.tf

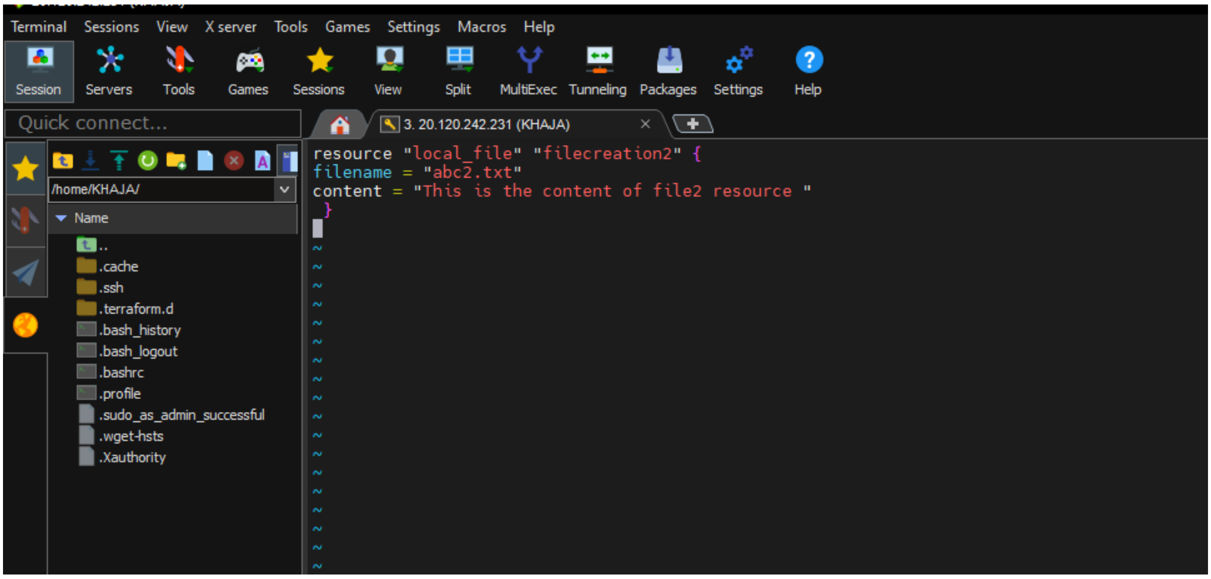
Create the second file content file2.tf and save it esc:wq

resource "local_file" "filecreation2" {

filename = "abc2.txt"

content = "This is the content of file2 resource "

}



Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
3. 20.120.242.231 (KHAJA)
/home/KHAJA/
Name
..
.cache
.ssh
.terraform.d
.bash_history
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.wget-hsts
.Xauthority

```
resource "local_file" "filecreation2" {
filename = "abc2.txt"
content = "This is the content of file2 resource "
 }
```

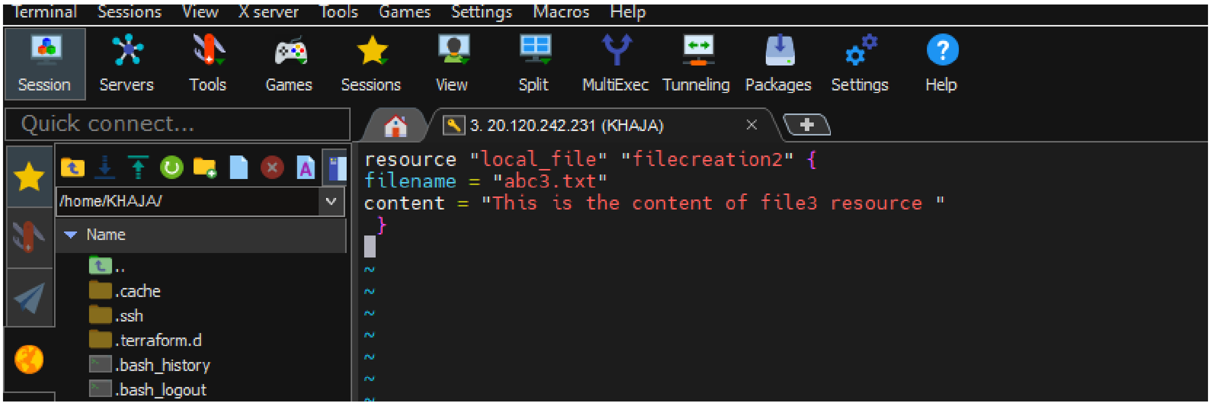
Create the third file content file3.tf and save it esc:wq

resource "local_file" "filecreation2" {

filename = "abc3.txt"

content = "This is the content of file3 resource "

}

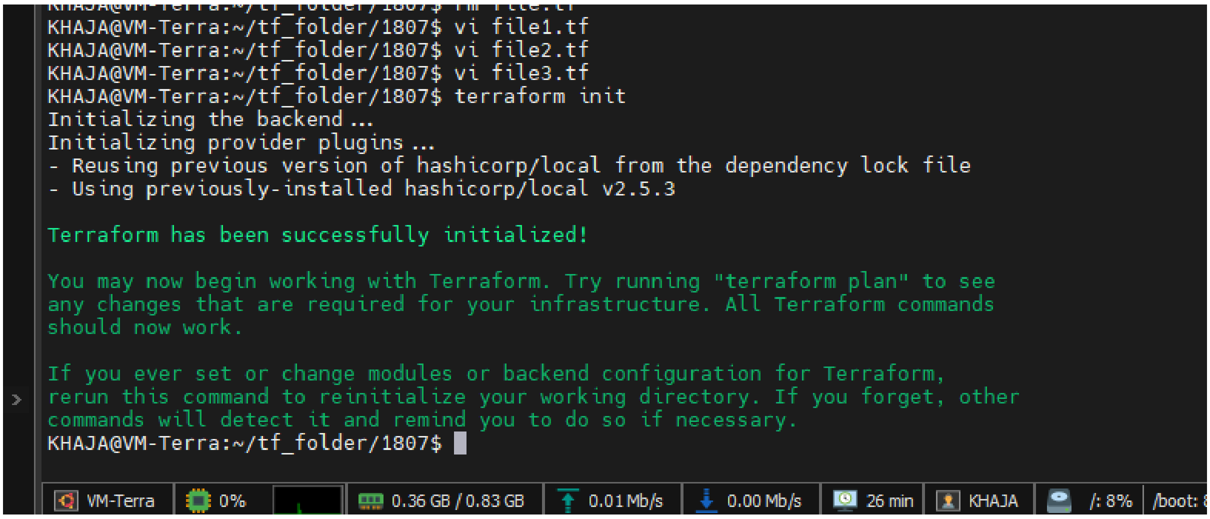


Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
3. 20.120.242.231 (KHAJA)
/home/KHAJA/
Name
..
.cache
.ssh
.terraform.d
.bash_history
.bash_logout

```
resource "local_file" "filecreation2" {
filename = "abc3.txt"
content = "This is the content of file3 resource "
 }
```

Run the command to Initialize Terraform

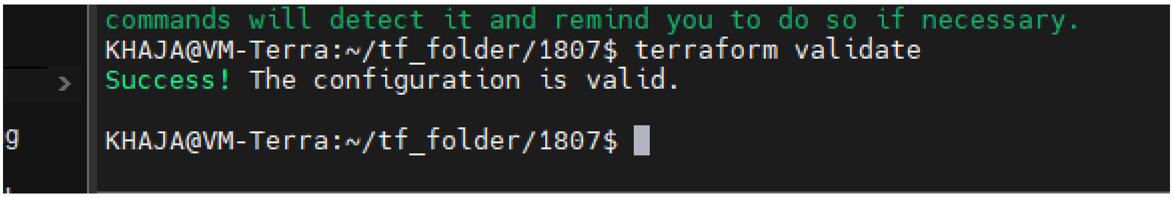
- ➢ terraform init



```
KHAJA@VM-Terra:~/tf_folder/1807$ rm file.tf
KHAJA@VM-Terra:~/tf_folder/1807$ vi file1.tf
KHAJA@VM-Terra:~/tf_folder/1807$ vi file2.tf
KHAJA@VM-Terra:~/tf_folder/1807$ vi file3.tf
KHAJA@VM-Terra:~/tf_folder/1807$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/local from the dependency lock file
- Using previously-installed hashicorp/local v2.5.3

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
KHAJA@VM-Terra:~/tf_folder/1807$
```

Will check for the validation

- ➢ terraform validate



```
commands will detect it and remind you to do so if necessary.
KHAJA@VM-Terra:~/tf_folder/1807$ terraform validate
Success! The configuration is valid.

KHAJA@VM-Terra:~/tf_folder/1807$
```

After that we need to run the following commands

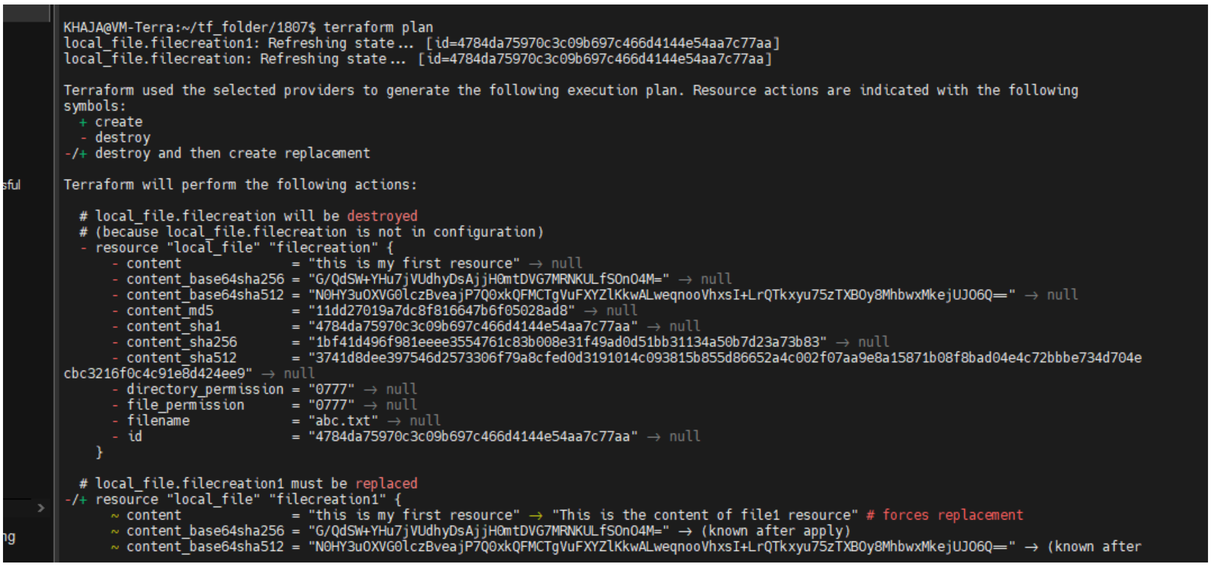
terraform plan

- it will show a preview or dry run.
- It shows like What Terraform wants to create, change, or delete.
- No changes are made to our real environment when running plan.

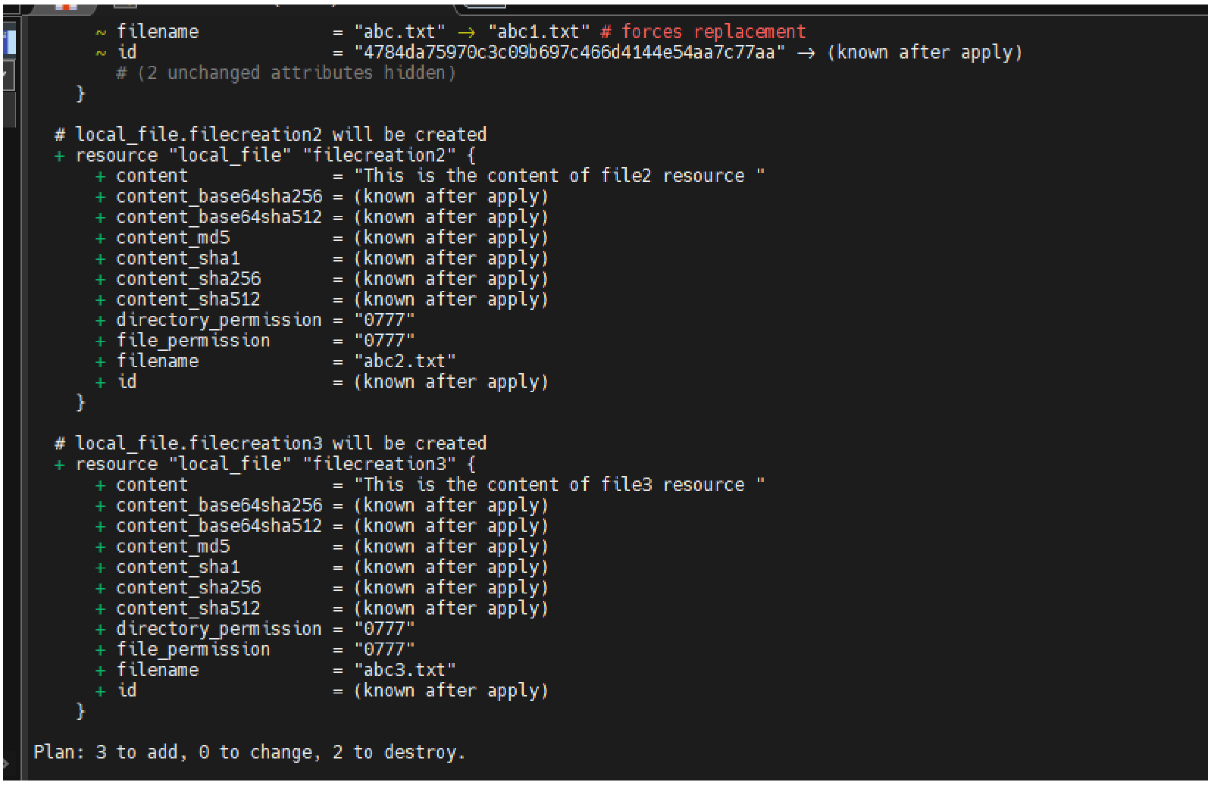
terraform apply

- This is the action step.
- Terraform uses our .tf files to actually: to Create, modify, or delete real resources (VMs, networks, etc.).
- We will usually get a confirmation prompt before changes happen like for proceeding yes/No

Run the command terraform plan



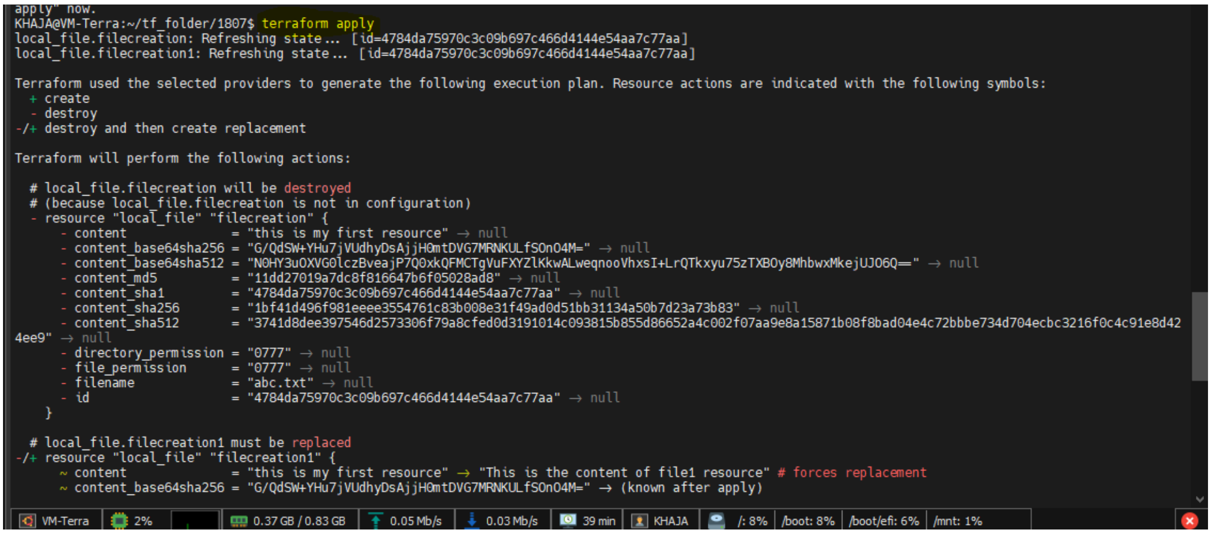
```
KHAJA@VM-Terra:~/tf_folder/1807$ terraform plan
local_file.filecreation1: Refreshing state... [id=4784da75970c3c09b697c466d4144e54aa7c77aa]
local_file.filecreation: Refreshing state... [id=4784da75970c3c09b697c466d4144e54aa7c77aa]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  + create
  - destroy
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # local_file.filecreation will be destroyed
  # (because local_file.filecreation is not in configuration)
  - resource "local_file" "filecreation" {
      - content              = "this is my first resource" -> null
      - content_base64sha256 = "G/QdSW+YHu7jVUdhyDsAjjH0mtDVG7MRNKULfSOn04M=" -> null
      - content_base64sha512 = "N0HY3uOXVG0lczBveajP7Q0xkQFMCTgVuFXYZlKkwALweqnooVhxsI+LrQTkxyu75zTXBOy8MhbwxMkejUJ06Q==" -> null
      - content_md5          = "11dd27019a7dc8f816647b6f05028ad8" -> null
      - content_sha1         = "4784da75970c3c09b697c466d4144e54aa7c77aa" -> null
      - content_sha256       = "1bf41d496f981eeee3554761c83b008e31f49ad0d51bb31134a50b7d23a73b83" -> null
      - content_sha512       = "3741d8dee397546d2573306f79a8cfed0d3191014c093815b855d86652a4c002f07aa9e8a15871b08f8bad04e4c72bbbe734d704e
cbc3216f0c4c91e8d424ee9" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "abc.txt" -> null
      - id                   = "4784da75970c3c09b697c466d4144e54aa7c77aa" -> null
    }

  # local_file.filecreation1 must be replaced
-/+ resource "local_file" "filecreation1" {
      ~ content              = "this is my first resource" -> "This is the content of file1 resource" # forces replacement
      ~ content_base64sha256 = "G/QdSW+YHu7jVUdhyDsAjjH0mtDVG7MRNKULfSOn04M=" -> (known after apply)
      ~ content_base64sha512 = "N0HY3uOXVG0lczBveajP7Q0xkQFMCTgVuFXYZlKkwALweqnooVhxsI+LrQTkxyu75zTXBOy8MhbwxMkejUJ06Q==" -> (known after
```



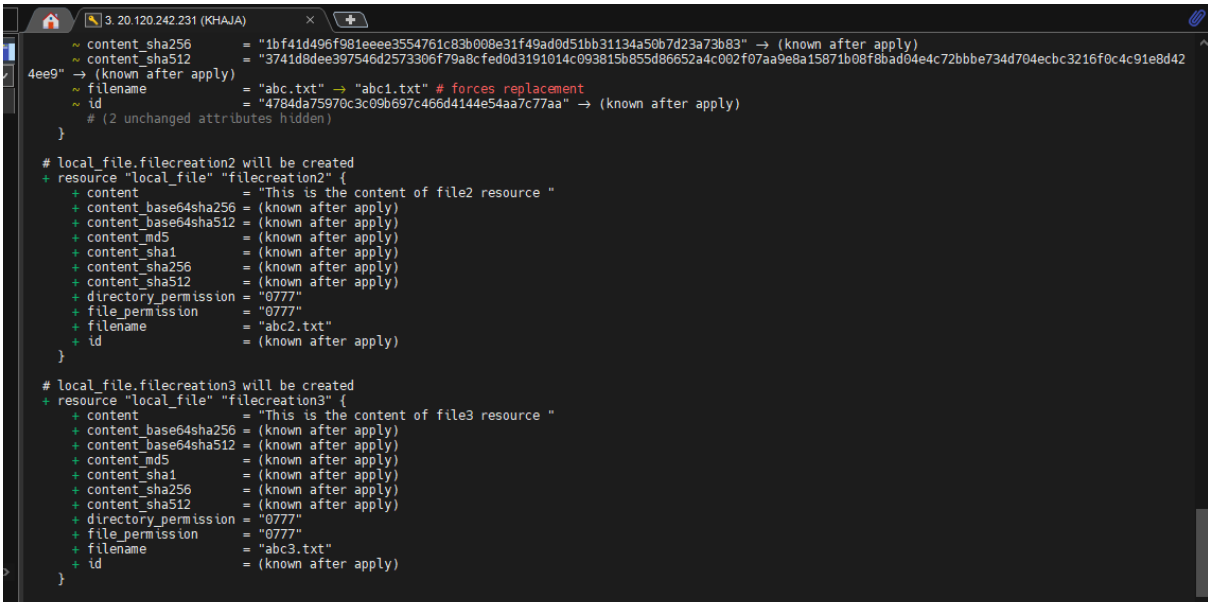
```
      ~ filename             = "abc.txt" -> "abc1.txt" # forces replacement
      ~ id                   = "4784da75970c3c09b697c466d4144e54aa7c77aa" -> (known after apply)
        # (2 unchanged attributes hidden)
    }

  # local_file.filecreation2 will be created
  + resource "local_file" "filecreation2" {
      + content              = "This is the content of file2 resource "
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "abc2.txt"
      + id                   = (known after apply)
    }

  # local_file.filecreation3 will be created
  + resource "local_file" "filecreation3" {
      + content              = "This is the content of file3 resource "
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "abc3.txt"
      + id                   = (known after apply)
    }

Plan: 3 to add, 0 to change, 2 to destroy.
```

Run the command terraform apply

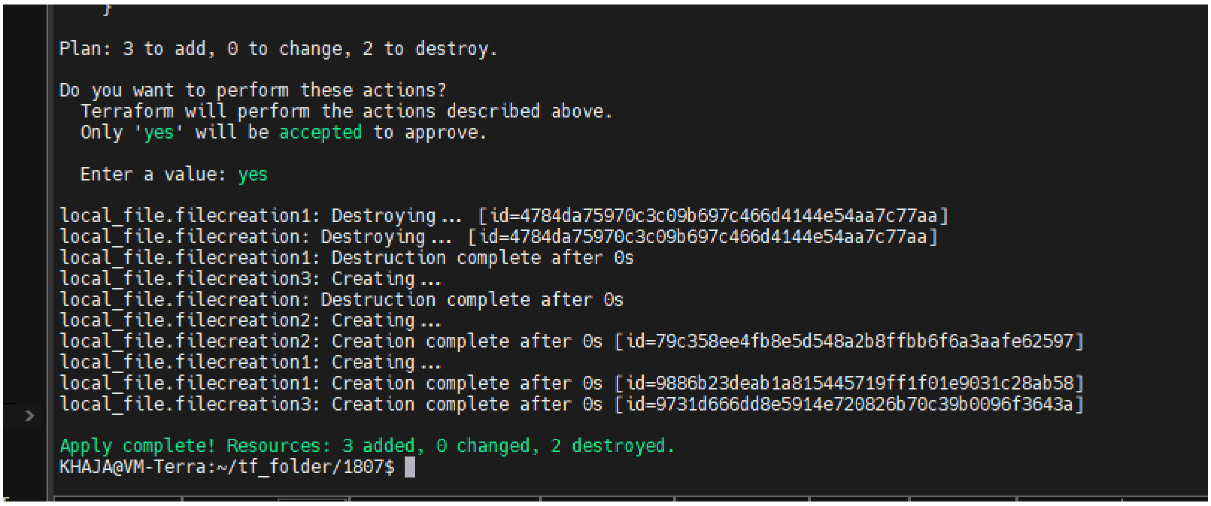
Terraform will show actions and prompt you for confirmation. Type yes when prompted.



```
apply" now.
KHAJA@VM-Terra:~/tf_folder/1807$ terraform apply
local_file.filecreation: Refreshing state... [id=4784da75970c3c09b697c466d4144e54aa7c77aa]
local_file.filecreation1: Refreshing state... [id=4784da75970c3c09b697c466d4144e54aa7c77aa]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create
  - destroy
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # local_file.filecreation will be destroyed
  # (because local_file.filecreation is not in configuration)
  - resource "local_file" "filecreation" {
      - content              = "this is my first resource" → null
      - content_base64sha256 = "G/QdSW+YHu7jVUdhyDsAjjH0mtDVG7MRNKULfSOn04M=" → null
      - content_base64sha512 = "N0HY3uOXVG0lczBveajP7Q0xkQFMCTgVuFXYZlKkwALweqnooVhxsI+LrQTkxyu75zTXBOy8MhbwxMkejUJO6Q==" → null
      - content_md5          = "11dd27019a7dc8f816647b6f05028ad8" → null
      - content_sha1         = "4784da75970c3c09b697c466d4144e54aa7c77aa" → null
      - content_sha256       = "1bf41d496f981eeee3554761c83b008e31f49ad0d51bb31134a50b7d23a73b83" → null
      - content_sha512       = "3741d8dee397546d2573306f79a8cfed0d3191014c093815b855d86652a4c002f07aa9e8a15871b08f8bad04e4c72bbbe734d704ecbc3216f0c4c91e8d42
4ee9" → null
      - directory_permission = "0777" → null
      - file_permission      = "0777" → null
      - filename             = "abc.txt" → null
      - id                   = "4784da75970c3c09b697c466d4144e54aa7c77aa" → null
    }

  # local_file.filecreation1 must be replaced
-/+ resource "local_file" "filecreation1" {
      ~ content              = "this is my first resource" → "This is the content of file1 resource" # forces replacement
      ~ content_base64sha256 = "G/QdSW+YHu7jVUdhyDsAjjH0mtDVG7MRNKULfSOn04M=" → (known after apply)
```



```
      ~ content_sha256       = "1bf41d496f981eeee3554761c83b008e31f49ad0d51bb31134a50b7d23a73b83" → (known after apply)
      ~ content_sha512       = "3741d8dee397546d2573306f79a8cfed0d3191014c093815b855d86652a4c002f07aa9e8a15871b08f8bad04e4c72bbbe734d704ecbc3216f0c4c91e8d42
4ee9" → (known after apply)
      ~ filename             = "abc.txt" → "abc1.txt" # forces replacement
      ~ id                   = "4784da75970c3c09b697c466d4144e54aa7c77aa" → (known after apply)
        # (2 unchanged attributes hidden)
    }

  # local_file.filecreation2 will be created
  + resource "local_file" "filecreation2" {
      + content              = "This is the content of file2 resource "
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "abc2.txt"
      + id                   = (known after apply)
    }

  # local_file.filecreation3 will be created
  + resource "local_file" "filecreation3" {
      + content              = "This is the content of file3 resource "
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "abc3.txt"
      + id                   = (known after apply)
    }
```



```
    }

Plan: 3 to add, 0 to change, 2 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.filecreation1: Destroying... [id=4784da75970c3c09b697c466d4144e54aa7c77aa]
local_file.filecreation: Destroying... [id=4784da75970c3c09b697c466d4144e54aa7c77aa]
local_file.filecreation1: Destruction complete after 0s
local_file.filecreation3: Creating...
local_file.filecreation: Destruction complete after 0s
local_file.filecreation2: Creating...
local_file.filecreation2: Creation complete after 0s [id=79c358ee4fb8e5d548a2b8ffbb6f6a3aafe62597]
local_file.filecreation1: Creating...
local_file.filecreation1: Creation complete after 0s [id=9886b23deab1a815445719ff1f01e9031c28ab58]
local_file.filecreation3: Creation complete after 0s [id=9731d666dd8e5914e720826b70c39b0096f3643a]

Apply complete! Resources: 3 added, 0 changed, 2 destroyed.
KHAJA@VM-Terra:~/tf_folder/1807$
```

3. **Create single .tf file with three resources in it**

Ans: We need to create a .tf files, with a three resources in it, here is a clear structure and example content for each file, using the [local_file resource from the local provider]

Create a single .tf file with:

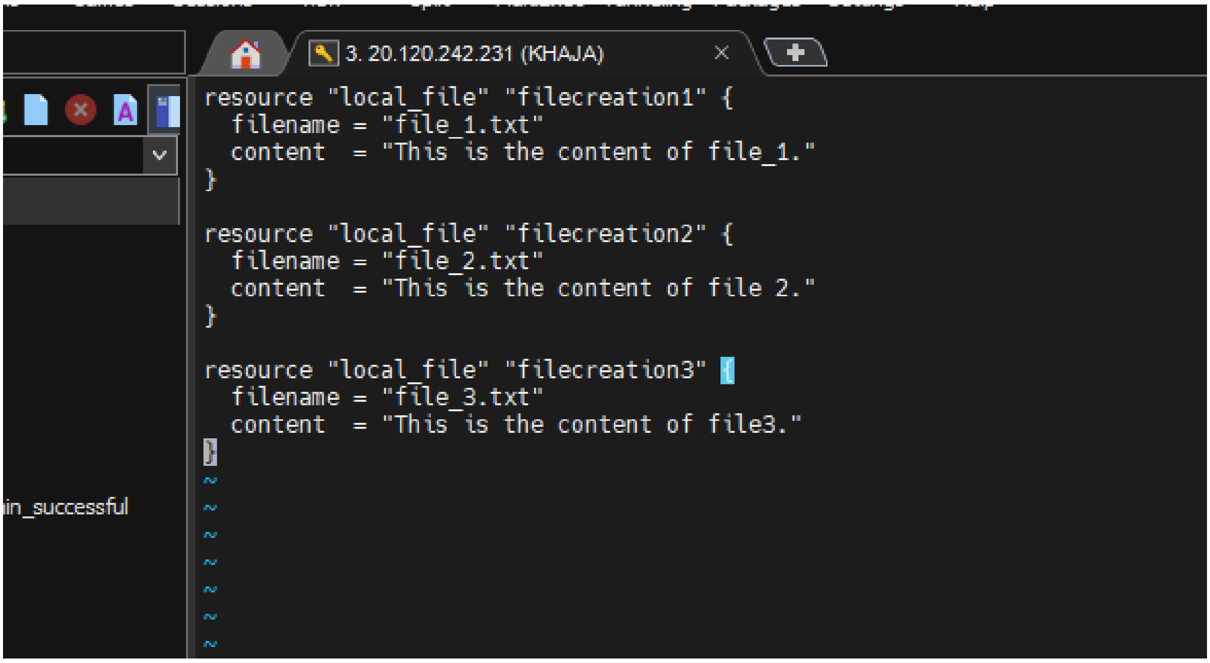
- **1 provider block**
- **3 resources:**
  - **abc1.txt**
  - **abc2.txt**
  - **abc3.txt**

Creating file

We create a file named main1.tf using vi file.tf

Enter the contents in file.tf as follows and save it using esc:wq

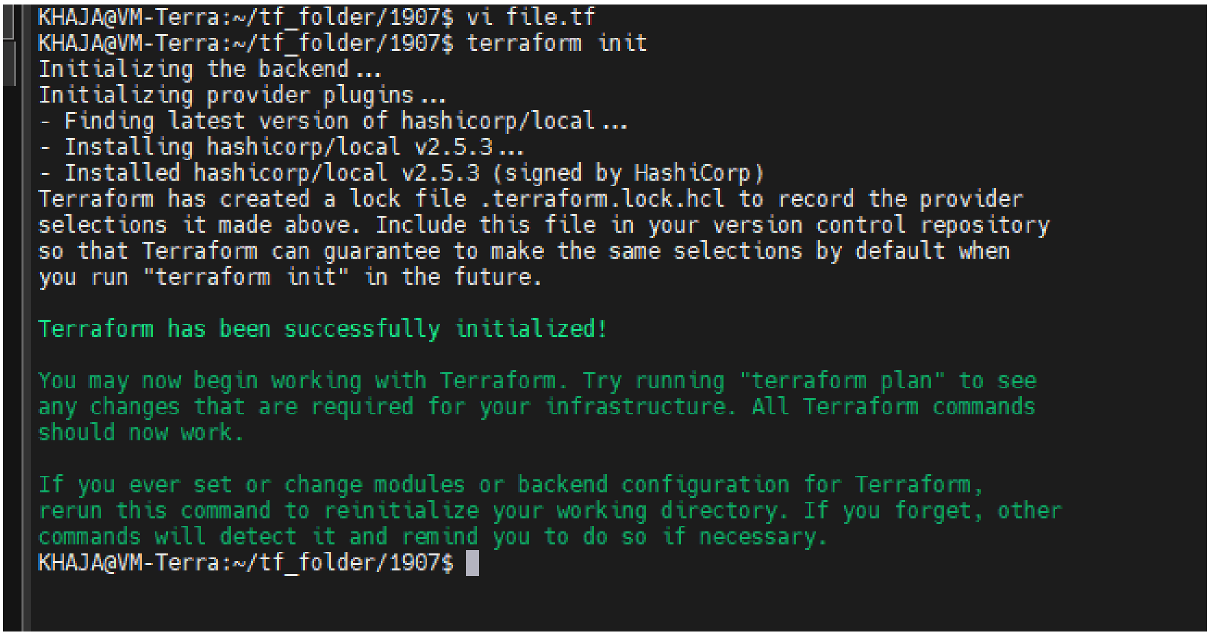
```
resource "local_file" "filecreation1" {
  filename = "file_1.txt"
  content  = "This is the content of file_1."
}

resource "local_file" "filecreation2" {
  filename = "file_2.txt"
  content  = "This is the content of file 2."
}

resource "local_file" "filecreation3" {
  filename = "file_3.txt"
  content  = "This is the content of file3."
}
```



```
resource "local_file" "filecreation1" {
  filename = "file_1.txt"
  content  = "This is the content of file_1."
}

resource "local_file" "filecreation2" {
  filename = "file_2.txt"
  content  = "This is the content of file 2."
}

resource "local_file" "filecreation3" {
  filename = "file_3.txt"
  content  = "This is the content of file3."
}
~
~
~
~
~
~
~
```

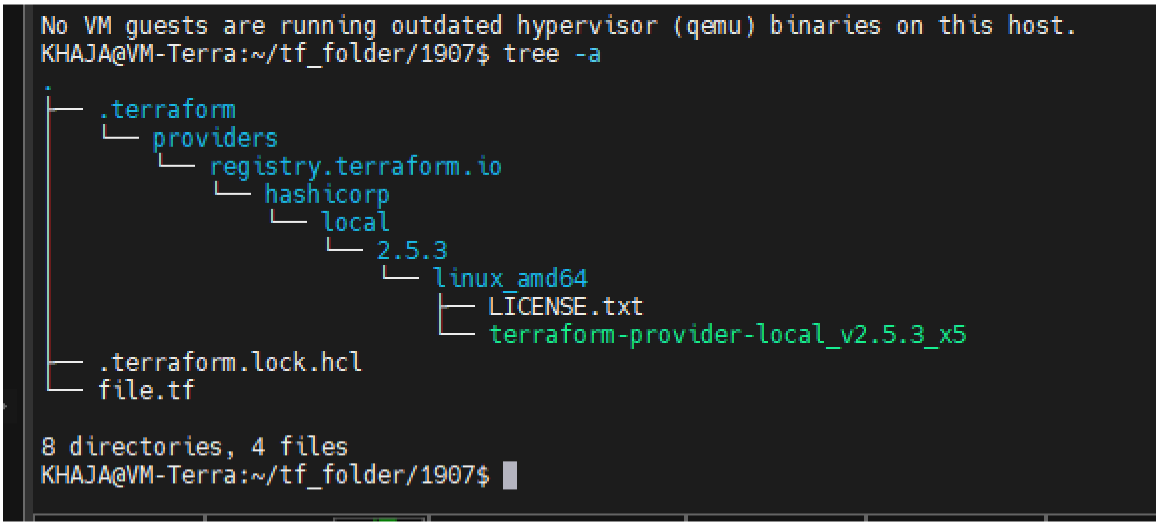
Initialize Terraform using command

- ➢ terraform init



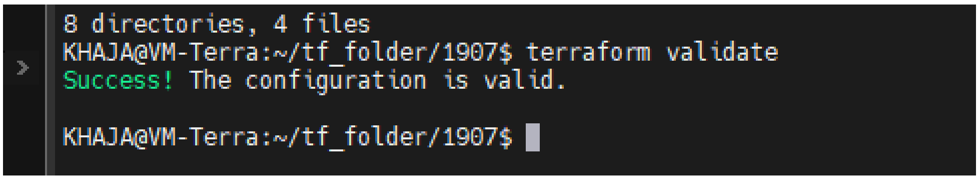
```
KHAJA@VM-Terra:~/tf_folder/1907$ vi file.tf
KHAJA@VM-Terra:~/tf_folder/1907$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
KHAJA@VM-Terra:~/tf_folder/1907$
```

Check the contents using tree -a



```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
KHAJA@VM-Terra:~/tf_folder/1907$ tree -a
.
├── .terraform
│   └── providers
│       └── registry.terraform.io
│           └── hashicorp
│               └── local
│                   └── 2.5.3
│                       └── linux_amd64
│                           ├── LICENSE.txt
│                           └── terraform-provider-local_v2.5.3_x5
├── .terraform.lock.hcl
└── file.tf

8 directories, 4 files
KHAJA@VM-Terra:~/tf_folder/1907$
```

Run the command terraform validate

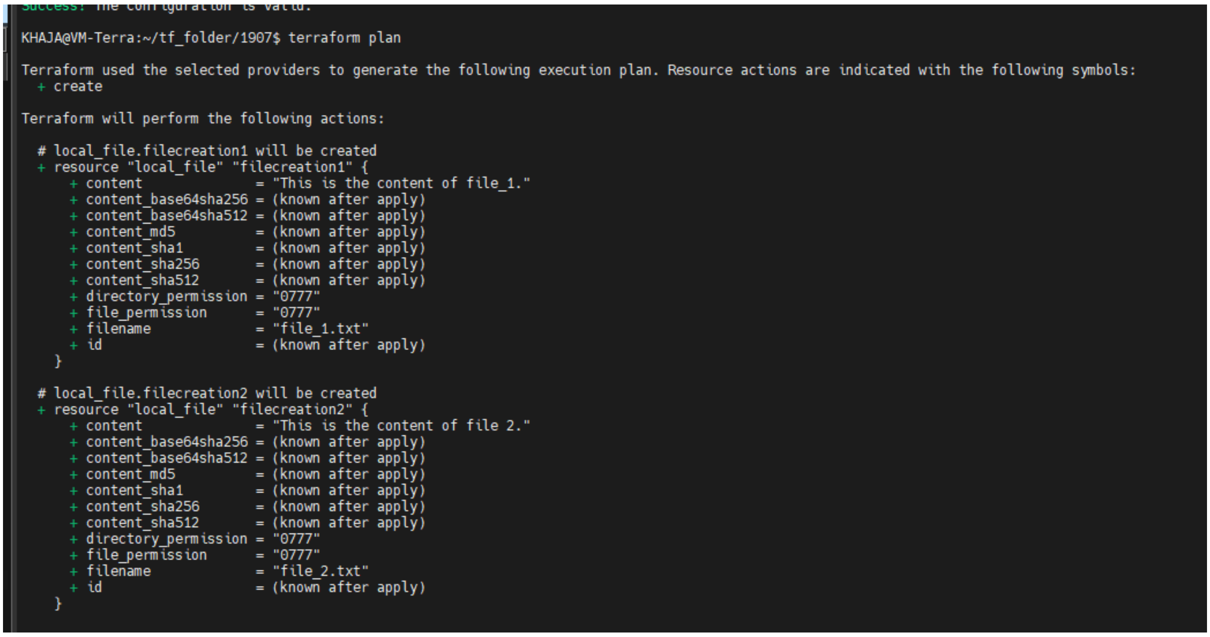


```
8 directories, 4 files
KHAJA@VM-Terra:~/tf_folder/1907$ terraform validate
Success! The configuration is valid.

KHAJA@VM-Terra:~/tf_folder/1907$
```

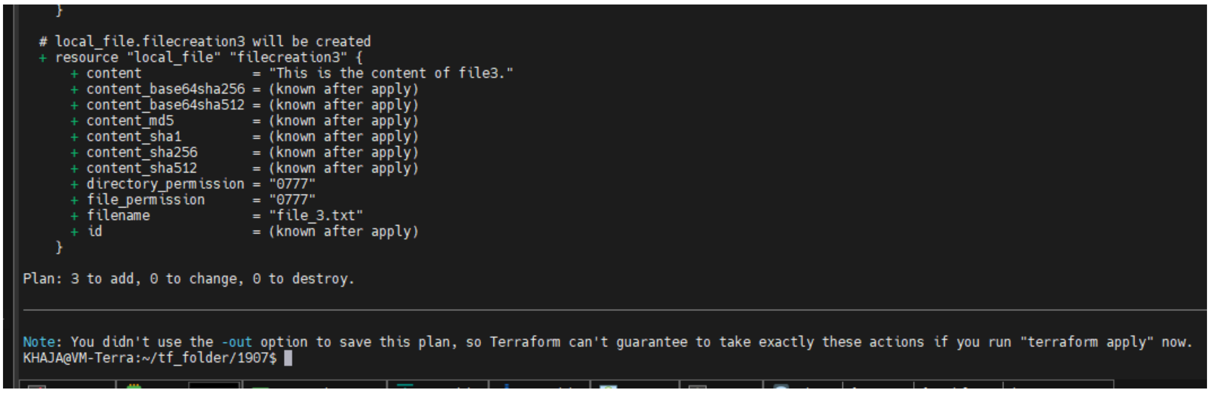
terraform plan

- it will show a preview or dry run.
- It shows like What Terraform wants to create, change, or delete.
- No changes are made to our real environment when running plan.

Run the command terraform plan



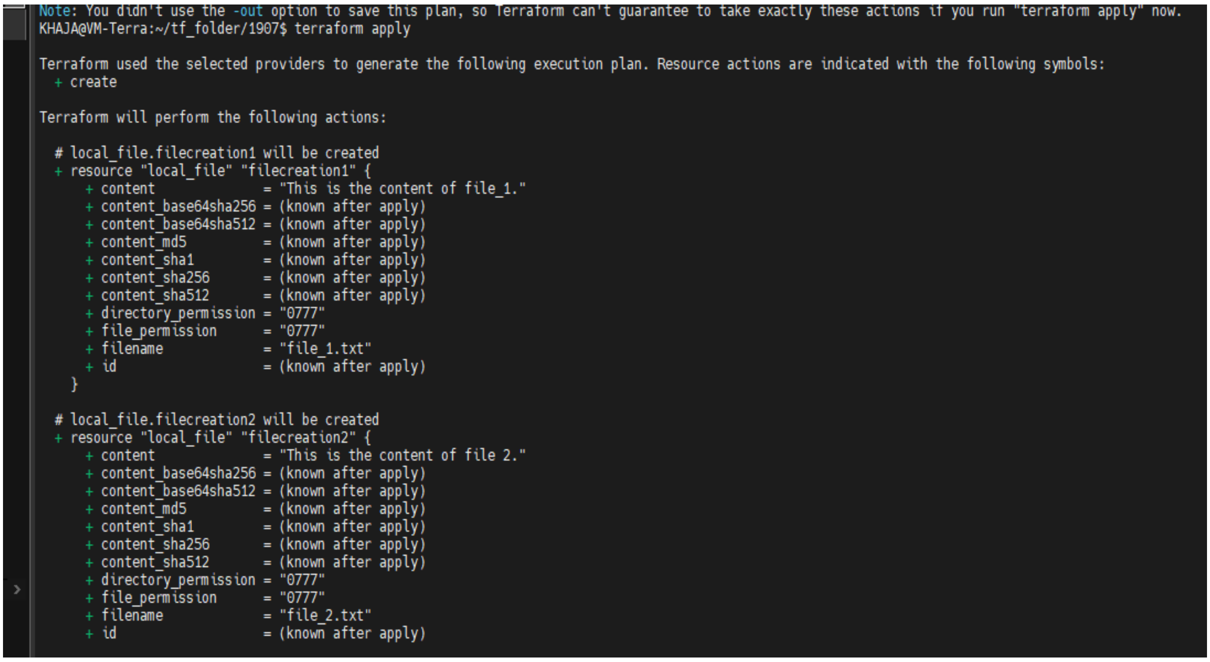
```
Success! The configuration is valid.

KHAJA@VM-Terra:~/tf_folder/1907$ terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # local_file.filecreation1 will be created
  + resource "local_file" "filecreation1" {
      + content              = "This is the content of file_1."
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "file_1.txt"
      + id                   = (known after apply)
    }

  # local_file.filecreation2 will be created
  + resource "local_file" "filecreation2" {
      + content              = "This is the content of file 2."
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "file_2.txt"
      + id                   = (known after apply)
    }
```



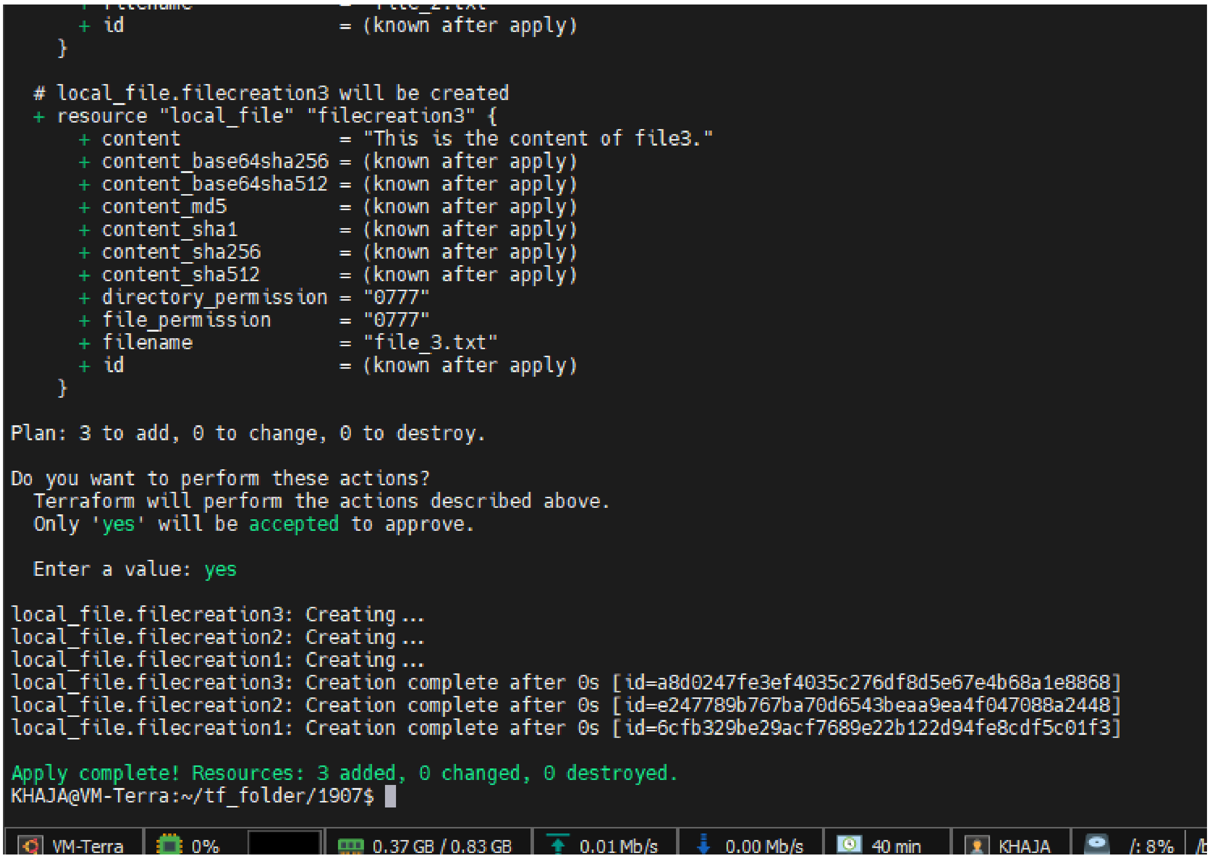
```
    }

  # local_file.filecreation3 will be created
  + resource "local_file" "filecreation3" {
      + content              = "This is the content of file3."
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "file_3.txt"
      + id                   = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
KHAJA@VM-Terra:~/tf_folder/1907$
```

terraform apply

- This is the action step.
- Terraform uses our .tf files to actually: to Create, modify, or delete real resources (VMs, networks, etc.).
- We will usually get a confirmation prompt before changes happen like for proceeding yes/No

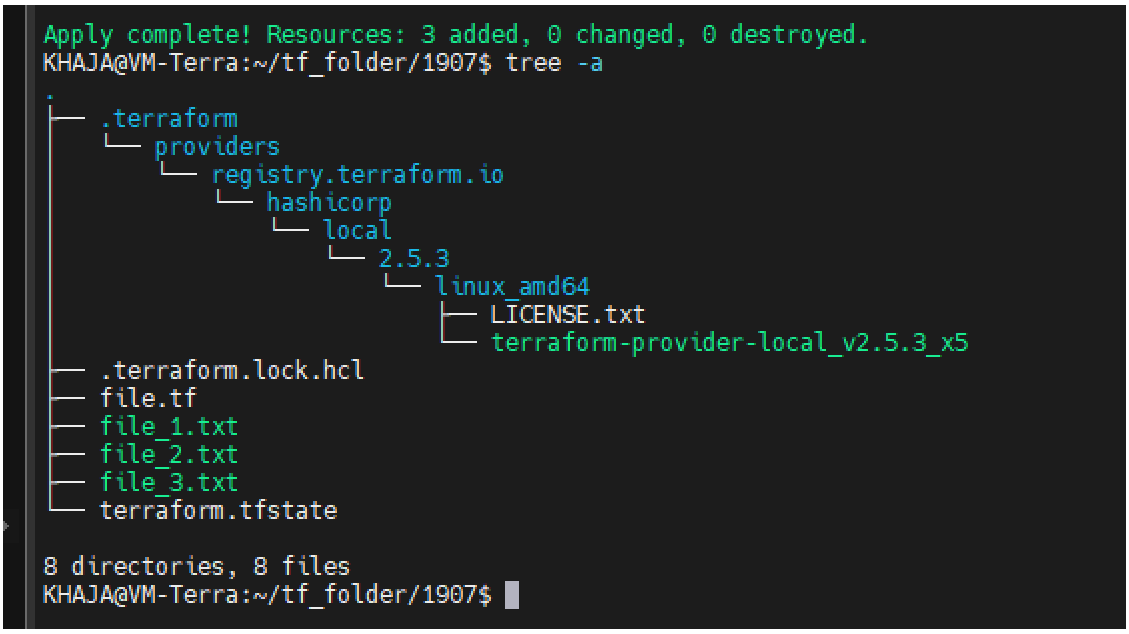


```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
KHAJA@VM-Terra:~/tf_folder/1907$ terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # local_file.filecreation1 will be created
  + resource "local_file" "filecreation1" {
      + content              = "This is the content of file_1."
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "file_1.txt"
      + id                   = (known after apply)
    }

  # local_file.filecreation2 will be created
  + resource "local_file" "filecreation2" {
      + content              = "This is the content of file 2."
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "file_2.txt"
      + id                   = (known after apply)
```



```
      + filename             = "file_2.txt"
      + id                   = (known after apply)
    }

  # local_file.filecreation3 will be created
  + resource "local_file" "filecreation3" {
      + content              = "This is the content of file3."
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "file_3.txt"
      + id                   = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.filecreation3: Creating...
local_file.filecreation2: Creating...
local_file.filecreation1: Creating...
local_file.filecreation3: Creation complete after 0s [id=a8d0247fe3ef4035c276df8d5e67e4b68a1e8868]
local_file.filecreation2: Creation complete after 0s [id=e247789b767ba70d6543beaa9ea4f047088a2448]
local_file.filecreation1: Creation complete after 0s [id=6cfb329be29acf7689e22b122d94fe8cdf5c01f3]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
KHAJA@VM-Terra:~/tf_folder/1907$
```

Check the contents of directory using tree -a



```
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
KHAJA@VM-Terra:~/tf_folder/1907$ tree -a
.
├── .terraform
│   └── providers
│       └── registry.terraform.io
│           └── hashicorp
│               └── local
│                   └── 2.5.3
│                       └── linux_amd64
│                           ├── LICENSE.txt
│                           └── terraform-provider-local_v2.5.3_x5
├── .terraform.lock.hcl
├── file.tf
├── file_1.txt
├── file_2.txt
├── file_3.txt
└── terraform.tfstate

8 directories, 8 files
KHAJA@VM-Terra:~/tf_folder/1907$
```